DOI：10. 11714/acta. snus. ZR20260118

基于改进型DDPG的AGV自主智能导航算法*

陈永灿[1]，黄晓[2]，梁海澄[1]，唐承佩[3]

1. 广东轻工职业技术大学智能制造与装备学院，广东 广州 510399
2. 中山大学微电子科学与技术学院，广东 珠海 519082
3. 中山大学智能工程学院，广东 深圳 518107

摘　要：提出了基于改进型深度确定性策略梯度（DDPG）的自动导引车（AGV）自主智能导航算法。首先，提出了基于环境风险的自适应探索策略，其中包含对动态障碍物轨迹的预判；其次，设计了多目标加权的奖励函数，指引AGV在避开障碍物且不越界的情况下尽快到达目的地；再次，基于风险系数-经验新鲜度双因素抽取经验，缩短训练周期，提升训练质量；最后，运行时的动作预演进一步降低碰撞和越界风险。为验证算法的性能，在不同障碍物数量、尺寸、移动速度条件下开展仿真实验。实验结果显示，基于改进型DDPG的AGV导航算法平均成功率达94.8%，平均超时率低于3%，平均碰撞率低于2.3%；训练周期和导航耗时亦短于对比算法，具有明显优势。

关键词：深度强化学习；DDPG；AGV；智能导航

中图分类号：TP242.6　　**文献标志码：**A　　**文章编号：**2097－0137(2026)05－0118－11

AGV autonomous intelligentn avigation algorithm based on improved DDPG

Chen Yongcan[1], *Huang Xiao*[2], *Liang Haicheng*[1], *Tang Chengpei*[3]

1. College of Intelligent Manufacturing and Equipment Engineering, Guangdong Industry Polytechnic University, Guangzhou 510399, China
2. School of Microelectronics Science and Technology, Sun Yat-sen University, Zhuhai 519082, China
3. School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen 518107, China

Abstract: An autonomous intelligent navigation algorithm for automatic guided vehicles (AGV) based on an improved deep deterministic policy gradient (DDPG) is proposed. First, an adaptive exploration strategy based on environmental risk is introduced, which includes predicting the trajectories of dynamic obstacles. Next, a multi-objective weighted reward function is designed to guide the AGV to reach its destination as quickly as possible while avoiding obstacles and staying within boundaries. Then, experiences are extracted using a dual-factor approach based on risk coefficients and experience freshness, which shortens the training cycle and improves training quality. Finally, action rehearsal during runtime further reduces the risk of collisions and boundary violations. To validate the algorithm's performance, simulation experiments were conducted under different numbers, sizes, and speeds of obstacles. The results showed that the AGV navigation algorithm based on the improved DDPG achieved an average success rate of 94.8%, an average timeout rate below 3%, and an average collision rate below 2.3%. Moreover, both the training cycles and navigation time were shorter than those of comparison algorithms, showing clear advantages.

Key words: deep reinforcement learning; DDPG; AGV; intelligent navigation

* **收稿日期：**2026-04-30　　**录用日期：**2026-06-11　　**网络首发日期：**2026-09-20
基金项目：广东省普通高校青年创新人才项目（2023KQNCX161）；
广东省自然科学基金（2024A151501214）；广州市重点研发计划（2025B03J0136）
作者简介：陈永灿（1988年生），男；**研究方向：**深度强化学习；E-mail：chenycan@alumni.sysu.edu.cn
通信作者：唐承佩（1978年生），男；**研究方向：**边缘智能；E-mail：tchengp@mail.sysu.edu.cn

自动引导车(AGV, automatic guided vehicles)广泛应用于仓储物流、车站码头等场所。传统AGV导航算法依赖人工部署的视觉标签，适用于固定环境，人力成本高。以同步定位与地图构建(SLAM, simultaneous localization and mapping)算法(Zhang et al., 2022)、A*算法(Liu et al., 2024)、APF(Wu, 2025)为代表的经典AGV导航算法摆脱了人工部署的限制，但在复杂环境下会出现定位不准、建图畸形等问题。其中，A*算法基于已知的静态地图规划路径，逻辑简单但无法动态避障；APF能实现局部避障但无法全局寻优。因此，经典AGV导航算法对复杂环境适应性差，难以预判动态障碍物轨迹，易陷入局部最优。

为解决上述问题，研究者提出基于深度强化学习(DRL, deep reinforcement learning)的AGV导航算法。王唯鉴(2023)采用图卷积网络聚合AGV和障碍物的关联特性，并利用深度Q学习评估动作价值，提出基于DRL的AGV动态避障方法。Shen et al.(2021)利用优势函数和熵训练深度神经网络(DNN, deep neural networks)，以解决分拣场景下的AGV路径规划难题。王贺等(2025)为避免AGV在避障时给行人带来不适，提出了基于DRL的端到端避障算法，通过YOLOv8获知行人位姿，并根据个人空间理论设计奖惩机制。Chen et al.(2023)以优化数据收集效率并降低电池消耗为目标，通过DRL解决了AGV的路径规划问题。

基于DRL的AGV导航算法提升了导航性能，但其性能由训练效果决定。复杂场景下的常规DRL训练周期长，训练质量不高，效果不稳定。为此，本研究提出了基于改进型深度确定性策略梯度(DDPG, deep deterministic policy gradient)的AGV自主智能导航算法。首先采用基于环境风险的自适应探索策略，并设计了多目标加权的奖励函数，以增强对复杂环境的适应能力；其次，基于风险系数-经验新鲜度双因素抽取经验，提高样本针对性、缩短训练周期、提升训练质量；最后，AGV运行时会预判动态障碍物的轨迹，结合自身决策实现动作预演，进一步降低碰撞和越界风险。研究还通过不同障碍物数量、尺寸和移动速度下的仿真实验，验证了基于改进型DDPG的AGV自主智能导航算法的有效性和鲁棒性。

1　AGV运动学模型

1.1　运动轨迹

为实现AGV的动态导航，本研究将AGV的路径按照较短的时间片($\Delta t \leqslant 1$ s)划分为多个路径片段。在实际运行中，AGV的转向依赖于对自身角速度和线速度的控制而非外力强制变轨，故每个时间片内的轨迹应为一条线段或圆弧。如图1所示，考虑到线段是为曲率为0的特殊圆弧，故采用圆弧刻画时间片Δt内的AGV运动轨迹。

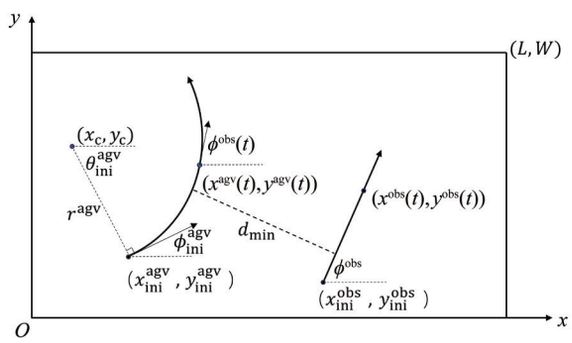

图1　单个时间片内的AGV和障碍物运动轨迹

Fig. 1　Motion trajectories of AGV and obstacle in a time slice

AGV通过雷达、红外等实时获知自身及障碍物状态。设AGV在某时间片段初始时的位置为$P_{\mathrm{ini}}^{\mathrm{agv}}\left(x_{\mathrm{ini}}^{\mathrm{agv}}, y_{\mathrm{ini}}^{\mathrm{agv}}\right)$，速度方向角为$\phi_{\mathrm{ini}}^{\mathrm{agv}}$，速度为$v^{\mathrm{agv}}$，角速度为$\omega^{\mathrm{agv}}$。当$\omega^{\mathrm{agv}}=0$时，AGV运动轨迹退化为1条线段；当$\omega^{\mathrm{agv}}\neq 0$时，由线速度和角速度可得圆弧半径

$$r^{\mathrm{agv}}=\frac{v^{\mathrm{agv}}}{\omega^{\mathrm{agv}}}. \tag{1}$$

圆心$\left(x_{\mathrm{c}}, y_{\mathrm{c}}\right)$位于起始位置速度方向的垂线上，速度方向的垂线方向角为$\phi_{\mathrm{ini}}^{\mathrm{agv}}+\frac{\pi}{2}\mathrm{sign}\left(\omega^{\mathrm{agv}}\right)$ rad，其中$\mathrm{sign}\left(\omega^{\mathrm{agv}}\right)$为符号函数。因此，圆心坐标为

$$\begin{cases} x_{\mathrm{c}}=x_{\mathrm{ini}}^{\mathrm{agv}}+r^{\mathrm{agv}}\cos\left(\phi_{\mathrm{ini}}^{\mathrm{agv}}+\pi\mathrm{sign}\left(\omega^{\mathrm{agv}}\right)/2\right), \\ y_{\mathrm{c}}=y_{\mathrm{ini}}^{\mathrm{agv}}+r^{\mathrm{agv}}\sin\left(\phi_{\mathrm{ini}}^{\mathrm{agv}}+\pi\mathrm{sign}\left(\omega^{\mathrm{agv}}\right)/2\right). \end{cases}$$

代入三角函数变换$\cos\left(\phi_{\mathrm{ini}}^{\mathrm{agv}}\pm\pi/2\right)=\mp\sin\left(\phi_{\mathrm{ini}}^{\mathrm{agv}}\right)$和$\sin\left(\phi_{\mathrm{ini}}^{\mathrm{agv}}\pm\pi/2\right)=\pm\cos\left(\phi_{\mathrm{ini}}^{\mathrm{agv}}\right)$，化简得：

$$\begin{cases} x_{\mathrm{c}}=x_{\mathrm{ini}}^{\mathrm{agv}}-r^{\mathrm{agv}}\mathrm{sign}\left(\omega^{\mathrm{agv}}\right)\sin\left(\phi_{\mathrm{ini}}^{\mathrm{agv}}\right), \\ y_{\mathrm{c}}=y_{\mathrm{ini}}^{\mathrm{agv}}+r^{\mathrm{agv}}\mathrm{sign}\left(\omega^{\mathrm{agv}}\right)\cos\left(\phi_{\mathrm{ini}}^{\mathrm{agv}}\right). \end{cases} \tag{2}$$

初始位置角$\theta_{\mathrm{ini}}^{\mathrm{agv}}$为从圆心指向起始位置$P_{\mathrm{ini}}^{\mathrm{agv}}$的有向线段与$x$轴正方向的夹角，则有

$$\theta_{\mathrm{ini}}^{\mathrm{agv}}=\arctan2\left(y_{\mathrm{ini}}^{\mathrm{agv}}-y_{\mathrm{c}}, x_{\mathrm{ini}}^{\mathrm{agv}}-x_{\mathrm{c}}\right), \tag{3}$$

其中$\arctan2\left(y, x\right)$函数可区分四个象限的角度，取

值范围为[$-\pi, \pi$]，可得 AGV 在 t 时刻的角位置为 $\theta^{\mathrm{agv}}(t)=\theta_{\mathrm{ini}}^{\mathrm{agv}}+\omega^{\mathrm{agv}}t$，方向角 $\phi^{\mathrm{agv}}(t)=\phi_{\mathrm{ini}}^{\mathrm{agv}}+\omega^{\mathrm{agv}}t$。则 AGV 在时刻 t 的位置 $P^{\mathrm{agv}}(t)$ 可表示为

$$\begin{cases} x^{\mathrm{agv}}(t)=x_c+r^{\mathrm{agv}}\cos\left(\theta_{\mathrm{ini}}^{\mathrm{agv}}+\omega^{\mathrm{agv}}t\right), \\ y^{\mathrm{agv}}(t)=y_c+r^{\mathrm{agv}}\sin\left(\theta_{\mathrm{ini}}^{\mathrm{agv}}+\omega^{\mathrm{agv}}t\right). \end{cases} \tag{4}$$

1.2 碰撞与越界判断

AGV 导航算法的本质是在避开障碍物和未越界的情况下尽快到达目的地，故需要有关于碰撞和越界的判断。

1.2.1 与场景内障碍物碰撞 AGV 通过感知系统获得自身的坐标、速度方向角以及障碍物的坐标和速度方向角、速度大小，并自主控制自身的速度和角速度。但 AGV 无法控制障碍物的角速度即无法干涉障碍物的转向，因此在时间段 Δt 内将障碍物的轨迹视为线段而非圆弧，如图 1 所示。设时间段开始时障碍物的坐标为 $\left(x_{\mathrm{ini}}^{\mathrm{obs}}, y_{\mathrm{ini}}^{\mathrm{obs}}\right)$，速度为 v^{obs}，速度方向角为 ϕ^{obs}，则障碍物在时刻 t 的位置 $P^{\mathrm{obs}}(t)$ 为

$$\begin{cases} x^{\mathrm{obs}}(t)=x_{\mathrm{ini}}^{\mathrm{obs}}+v^{\mathrm{obs}}\cos\left(\phi^{\mathrm{obs}}\right)t, \\ y^{\mathrm{obs}}(t)=y_{\mathrm{ini}}^{\mathrm{obs}}+v^{\mathrm{obs}}\sin\left(\phi^{\mathrm{obs}}\right)t. \end{cases} \tag{5}$$

判断 AGV 是否与障碍物发生碰撞是考察时间段 Δt 内，AGV 与障碍物的最小欧式距离是否小于或等于碰撞阈值。二者的最小欧式距离为

$$\begin{aligned} d_{\min} &= \min_{t\in[0,\Delta t]} d\left[P^{\mathrm{agv}}(t), P^{\mathrm{obs}}(t)\right] \\ &= \min_{t\in[0,\Delta t]} \left\| P^{\mathrm{agv}}(t)-P^{\mathrm{obs}}(t)\right\|_2. \end{aligned} \tag{6}$$

在仿真时，使用 Python 的 minimize_scalar 工具在[$0, \Delta t$]内求最小值。若场景内有多个障碍物，只需求 AGV 与每个障碍物的距离，然后从中取最小值。为接近实际，本文将 AGV 和障碍物视为有尺寸的圆，上述坐标即为圆心坐标。设 AGV 半径为 R^{agv}，障碍物半径为 R^{obs}，显然碰撞阈值为 $d_{\mathrm{col}}=R^{\mathrm{agv}}+R^{\mathrm{obs}}$，若 $d_{\min}\leqslant d_{\mathrm{col}}$，则表示 AGV 与场景内障碍物发生碰撞，反之未碰撞。

1.2.2 越界 边界一般为矩形框架的墙壁或边界线等，设场景的长度为 L，宽度为 W，如图 1 所示。为使 AGV 不越界，其中心位置需与场景边界保持不小于自身半径的安全距离，故其越界的判定标准为

$$x^{\mathrm{agv}}(t)\leqslant R^{\mathrm{agv}} \vee x^{\mathrm{agv}}(t)\geqslant L-R^{\mathrm{agv}} \vee y^{\mathrm{agv}}(t)\leqslant R^{\mathrm{agv}} \vee y^{\mathrm{agv}}(t)\geqslant W-R^{\mathrm{agv}},$$

否则，未越界。

2 基于改进型 DDPG 的 AGV 导航算法

考虑运动学约束与避障的 AGV 轨迹导航是高维状态空间中的非凸优化问题。强化学习（RL）的目标是最大化累积折扣奖励，而非仅关注当前奖励，故更容易避免陷入局部最优。DRL 具备端到端的学习和决策能力，不依赖精确环境模型。虽然 DRL 以 DNN 为基础，可处理复杂高维状态，但其性能取决于训练效果。在处理动态复杂场景中的 AGV 导航任务时，其训练周期长、策略收敛慢、训练结果不稳定。在各类 DRL 算法中，DDPG 针对连续动作空间设计（Lillicrap et al., 2019），适用于 AGV 连续动作控制与导航任务。本文以常规 DDPG 为基础，通过优化其网络架构、训练机制及推理模式，提出了基于改进型 DDPG 的 AGV 智能导航算法。

2.1 马尔可夫决策过程与深度强化学习

马尔可夫决策（MDP）是 RL 的基础数学模型，定义为 $M=(S, A, P, R, \gamma)$。其中 S 为状态空间，描述环境所有可能的状态；A 为动作空间，表征智能体所有可执行的动作；$P: S\times A\times S\rightarrow[0,1]$ 为状态转移概率，指智能体在状态 $s\in S$ 下执行动作 $a\in A$ 后，转移到状态 $s'\in S$ 的概率；$R: S\times A\times S\rightarrow\mathbb{R}$ 为即时奖励，用于评价动作优劣；$\gamma\in[0,1]$ 为折扣因子，用于平衡即时奖励与累积奖励（Gao et al., 2022）。

RL 的目标是求解最优策略 $\pi^*: S\rightarrow A$，以最大化累积折扣奖励，该奖励被定义为 $G_t=\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}$。为求解最优策略，引入状态价值函数 $V^{\pi}(s)$ 和动作价值函数 $Q^{\pi}(s,a)$，两者分别表示在状态 s 下遵循策略 π 所能获得的长期累积奖励期望和在状态 s 下执行动作 a 后再遵循策略 π 所能获得的长期累积奖励期望（Sutton et al., 2018）。

传统 RL 算法通常采用表格存储 $V^{\pi}(s)$ 和 $Q^{\pi}(s,a)$，在 AGV 导航等复杂高维状态空间下，表格存储会出现“维数灾难”。DRL 结合深度学习（DL, deep learning）的特征提取能力与 RL 的决策能力，采用 DNN 代替表格来拟合价值函数，突破了高维状态空间下求解问题的瓶颈（Arulkumaran et al., 2017; Wang et al., 2024）。

深度 Q 网络（DQN, deep Q-network）是 DRL 的经典算法，专为离散动作空间设计，其核心是用 DNN 拟合动作价值函数 $Q_{\theta}(s,a)$，其中 θ 为网络参数。为提升训练稳定性，DQN 引入经验回放机制和目标网络，经验回放将智能体与环境交互产生的经验 (s,a,r,s') 存储于回放缓冲区，训练时随机采样以避免样本相关性；目标网络参数 θ' 每隔一段时间从主网络参数 θ 复制更新（Haffz, 2022）。DQN 的损

失函数定义为

$$\mathcal{L}(\theta)=E\left[\left(r+\gamma\max_{a'\in A}Q_{\theta'}(s',a')-Q_{\theta}(s,a)\right)^2\right].$$

DQN仅适于离散动作空间，但AGV的转向角、行驶速度等动作多为连续值；若采用DQN，则需离散化且效果欠佳。

2.2　改进型DDPG算法的架构

本文所述的改进型DDPG框架如图2所示。Actor网络$\mu_\theta(s)$的输入是状态s，输出是动作a，其核心目标是学习最优策略$\mu_\theta^*(s)$。Critic网络$Q_\phi(s,a)$的目标是拟合Q值函数以评估“状态-动作对”的价值。目标Actor网络$\mu_{\theta'}(s')$/目标Critic网络$Q_{\phi'}(s',a')$的结构与Actor网络/Critic网络相同，但参数更新更低频，以避免训练时的震荡。经验回放池存储智能体与环境交互产生的经验$(s,a,r,s',\text{done},\tau)$，其中的元素分别表示状态、动作、奖励、下一状态、结束标志（Lillicrap et al.，2019）、经验入池时的全局步数。AGV通过感知系统获知障碍物坐标、速度大小和方向角以及自身的坐标和方向角，并据此输出控制指令即自身线速度大小和角速度，故s可表示为

$$s=\left[\left\{x_i^{\text{obs}},y_i^{\text{obs}},v_i^{\text{obs}},\phi_i^{\text{obs}}\right\}_1^M,\ x^{\text{agv}},\ y^{\text{agv}},\ \phi^{\text{agv}},\ d^{\text{dis\_agv}},\ \phi^{\text{dis\_agv}},\ d^{\text{b\_a}},\ \text{DIR}^{\text{bou}}\right],\tag{7}$$

其中$(x_i^{\text{obs}},y_i^{\text{obs}})$为第$i$个障碍物的坐标，$v_i^{\text{obs}}$和$\phi_i^{\text{obs}}$为其线速度大小和方向角，$M$表示障碍物数量。$(x^{\text{agv}},y^{\text{agv}})$为AGV的坐标，$\phi^{\text{agv}}$是AGV的方向角。AGV的线速度大小和角速度是控制信息，处在动作空间中。为指引AGV向目的地靠近，在状态空间中设置了AGV到达目的地的距离$d^{\text{dis_agv}}$和方向角$\phi^{\text{dis_agv}}$。此外，为标识越界风险并指导智能决策，状态空间还包含了AGV到最近边界的距离$d^{\text{b_a}}$及边界属性DIR^{bou}，其中$\text{DIR}^{\text{bou}}=1$、2、3、4分别表示左、右、下、上边界。

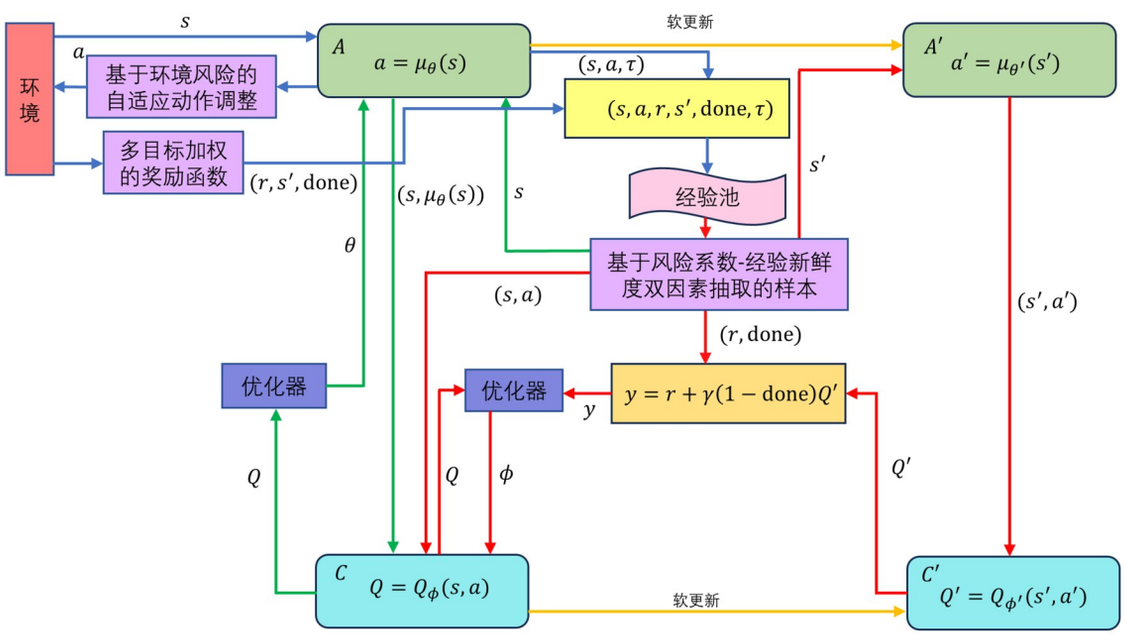

图2　改进型DDPG的框架

Fig. 2　Improved DDPG framework

经过Actor网络输出的动作a包括AGV的线速度大小和角速度，即$a=[v^{\text{agv}},\omega^{\text{agv}}]$。为提高动作质量，本研究在Actor输出的原始动作上又基于环境风险进行自适应动作调整，其逻辑如图3所示。设边界预警距离为d_{bou}，若$d^{\text{b_a}}\leqslant d_{\text{bou}}$，则表示接近边界；根据状态空间中的$\text{DIR}^{\text{bou}}$可知最近的边界方向及对应的远离方向，如$\text{DIR}^{\text{bou}}=1$表示靠近左边界、远离方向为$x$轴正方向。判断AGV当前朝向的左右侧对应的绝对方向，并计算二者与远离方向的夹角，选较小者作为转向方向，方能使得AGV朝远离边界的方向转向。若AGV左转，则$\omega^{\text{agv}}=\pi/2\ \text{rad}\cdot\text{s}^{-1}$，否则$\omega^{\text{agv}}=-\pi/2\ \text{rad}\cdot\text{s}^{-1}$。此外，当AGV接近边界时，应适当减速以降低AGV越界的概率；否则，应加速以尽快到达目的地。具体调速逻辑为

$$v^{\mathrm{agv}} \leftarrow \begin{cases} \varepsilon v^{\mathrm{agv}}, & d^{\mathrm{b\_a}} > d_{\mathrm{bou}}, \\ 0, & d^{\mathrm{b\_a}} \leqslant R^{\mathrm{agv}}, \\ \dfrac{d^{\mathrm{b\_a}} - R^{\mathrm{agv}}}{d_{\mathrm{bou}} - R^{\mathrm{agv}}} v^{\mathrm{agv}}, & \text{others}, \end{cases} \tag{8}$$

其中 $\varepsilon \in (1, 2]$ 表示速度增益系数。令 d_{warn} 表示 AGV 与障碍物的碰撞预警阈值，一般是在 d_{col} 的基础上加上缓冲距离 d_{buf}，即 $d_{\mathrm{warn}} = d_{\mathrm{col}} + d_{\mathrm{buf}}$。设 K 表征 AGV 和场景内障碍物的碰撞风险，定义为

$$K = \begin{cases} 0, & d_{\mathrm{min}} > d_{\mathrm{warn}}; \\ 1 - \dfrac{d_{\mathrm{min}} - d_{\mathrm{col}}}{d_{\mathrm{warn}} - d_{\mathrm{col}}}, & d_{\mathrm{col}} < d_{\mathrm{min}} \leqslant d_{\mathrm{warn}}; \\ 1, & d_{\mathrm{min}} \leqslant d_{\mathrm{col}}. \end{cases} \tag{9}$$

$K = 0$ 时表示无风险，此时 AGV 线速度加快，期望尽快到达目的地；有风险时，AGV 按照衰减系数 $(1 - K)$ 减速直至停止，即：

$$v^{\mathrm{agv}} \leftarrow \begin{cases} \varepsilon v^{\mathrm{agv}}, & K = 0; \\ (1 - K) v^{\mathrm{agv}}, & 0 < K < 1; \\ 0, & K = 1. \end{cases} \tag{10}$$

因在计算 d_{min} 时涉及对动态障碍物轨迹的预判，且考虑了场景内的全部障碍物，故采用自适应调整策略选择 AGV 动作，可使改进型 DDPG 以更大概率产生高质量经验，进而加快收敛速度，改善训练效果，更好地适应复杂环境。

2.3 改进型 DDPG 算法的训练

为使 Actor 学会接近并达到目的地、避障、防越界等行为，本研究将接近目的地奖励 r_{app}、到达目的地奖励 r_{dest}、接近（含碰撞）障碍物惩罚 r_{obs}、接近（含越过）边界的惩罚 r_{bound} 以加权形式纳入奖励函数中，即：

$$r = \lambda_a r_{\mathrm{app}} + \lambda_d r_{\mathrm{dest}} + \lambda_o r_{\mathrm{obs}} + \lambda_b r_{\mathrm{bound}}, \tag{11}$$

其中 r_{app} 依据 s 和 s' 状态下 AGV 到目的地的距离减小情况决定，有 $r_{\mathrm{app}} = d_s\left(P^{\mathrm{agv}}, P^{\mathrm{dest}}\right) - d_{s'}\left(P^{\mathrm{agv}}, P^{\mathrm{dest}}\right)$.

考虑到 AGV 是有尺寸的刚体，目的地也是有尺寸的区域，故设置目的地区域是以 P^{dest} 为圆心、以 d^{dest} 为半径的圆；若 $d_{s'}\left(P^{\mathrm{agv}}, P^{\mathrm{dest}}\right) \leqslant d^{\mathrm{dest}}$ 即表示 AGV 到达目的地，此时 $r_{\mathrm{dest}} = 1$，否则 $r_{\mathrm{dest}} = 0$。由于到达目的地应获得比接近目的地更多的奖励，故设置 $\lambda_a = 50$、$\lambda_d = 100$。

根据 AGV 接近障碍物和边界的程度可得：

$$r_{\mathrm{obs}} = \begin{cases} -1, & d_{\mathrm{min}} \leqslant d_{\mathrm{col}}; \\ -K, & d_{\mathrm{col}} < d_{\mathrm{min}} < d_{\mathrm{warn}}; \\ 0, & d_{\mathrm{min}} \geqslant d_{\mathrm{warn}}; \end{cases}$$

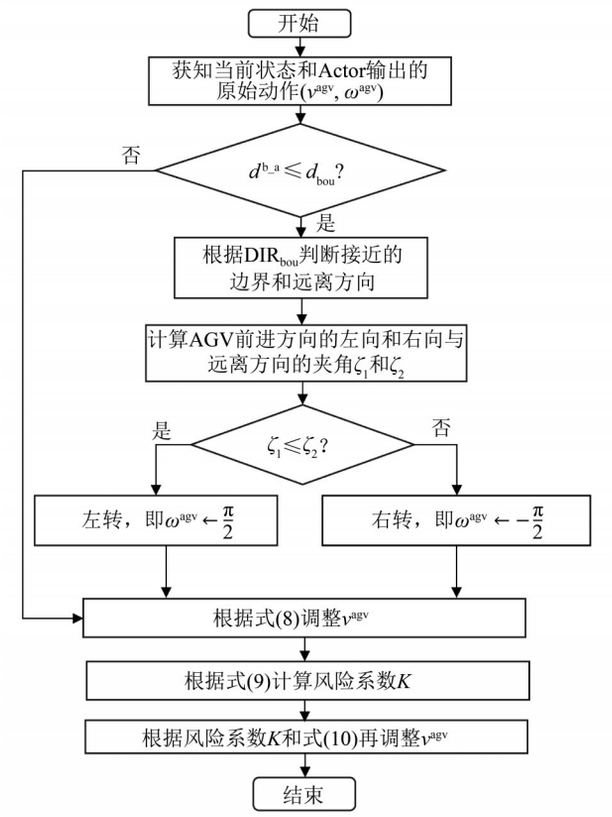

图 3 基于环境风险的自适应动作调整逻辑

Fig. 3 Adaptive action adjustment logic based on environmental risk

$$r_{\mathrm{bound}} = \begin{cases} 0, & d^{\mathrm{b\_a}} > d_{\mathrm{bou}}; \\ -1, & d^{\mathrm{b\_a}} \leqslant R^{\mathrm{agv}}; \\ -\dfrac{d_{\mathrm{bou}} - d^{\mathrm{b\_a}}}{d^{\mathrm{bou}} - R^{\mathrm{agv}}}, & \text{others}. \end{cases}$$

式（11）中的 $\lambda_o = 100$，$\lambda_b = 50$。若 AGV 到达目的地，或发生碰撞越界，或者步数达到最大值，则本轮训练终止；此时 done = 1，否则 done = 0。每训练 1 步，就将 1 条经验存入经验池，上述流程如图 2 中蓝色箭头线所示。当池中经验条数达到阈值时，本文基于风险系数-经验新鲜度双因素从经验池中抽取 N 条经验用于训练。风险系数 K 见式（8），经验新鲜度表征经验入池的时间远近。每条经验入池时，以经验中的 1 个字段 τ 标记其入池时的全局步数。设在进行经验抽取时的全局步数为 τ'，则步差 $\Delta\tau = \tau' - \tau$。设定经验池的最大有效时间窗 T_{max}，超过该步数的经验视为“完全不新鲜”，则经验新鲜度 $F = \max(0, 1 - \Delta\tau / T_{\mathrm{max}})$.

先为风险系数和经验新鲜度分配权重系数，再计算每个样本的综合权重 $w_{\mathrm{total}} = \lambda_K K + \lambda_F F$，对池中所有经验的权重做归一化处理，得到每条经验的采样概率：

$$p_i = w_{\text{total},i} \Big/ \sum_{j=1}^{U} w_{\text{total},j},$$

其中U表示池中经验条数。然后采用加权随机采样法，从经验池中抽取N个样本，显然权重越高，被抽中的概率越大。基于风险系数-经验新鲜度双因素的采样方法，本质上是提高风险大且新鲜的经验被抽中的概率，即及时采用新鲜样本训练DDPG，使之尽快学会风险应对策略，提高对复杂环境的适应能力，缩短训练周期、提高训练质量。

训练Critic网络的流程如图2中红色箭头线所示，依据抽取的样本，计算其中第i条经验$(s_i, a_i, r_i, s_i', \text{done}_i, \tau_i)$的目标$Q$值：

$$y_i = r_i + \gamma(1 - \text{done}_i)Q_{\phi'}(s_i', \mu_{\theta'}(s_i')),$$

其中$i \in [0, N-1]$。进而以最小化y_i与Critic输出$Q_\phi(s_i, a_i)$的均方误差为目标，通过梯度下降更新Critic的网络参数，即：

$$\phi \leftarrow \phi - \beta\nabla_\phi\left[\frac{1}{N}\sum_{i=1}^{N}\left(y_i - Q_\phi(s_i, a_i)\right)^2\right],$$

其中β是Critic网络的学习率。训练Actor网络的流程如图2中绿色箭头线所示，本质上是以“最大化Critic评估的Q值”为目标，通过策略梯度更新Actor网络参数θ：

$$\theta \leftarrow \theta + \alpha\nabla_\theta J(\theta),$$

其中

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\nabla_a Q_\phi(s_i, a)\Big|_{a=\mu_\theta(s_i)} \cdot \nabla_\theta \mu_\theta(s_i),$$

α是Actor网络的学习率。最后，软更新目标Actor和Critic网络的参数，具体策略为

$$\theta' \leftarrow \eta\theta + (1-\eta)\theta', \quad \phi' \leftarrow \eta\phi + (1-\eta)\phi',$$

其中η为软更新系数，该部分流程如图2中橙色箭头所示。

2.4　改进型DDPG算法的运行

改进型DDPG经过训练之后，其Actor网络输出的动作可以较大概率接近最优解，为进一步提升AGV运行的安全性和效率，本研究在DDPG的运行环节将推理输出和动作预演相结合，根据对复杂环境的预判进一步优化动作指令。首先，AGV通过感知获取环境状态s，经过推理模式的Actor网络输出动作a。AGV在已知s和a的情况下，可预演未来一个较短时间片Δt内是否会接近或撞击障碍物，是否会接近或越过界限，即通过式(1)~(6)计算得AGV和障碍物的最小距离$d_{\min}$，并通过如下公式获知AGV到边界的最小距离：

$$d^{b\_a} = \min_{t\in[0,\Delta t]}\left\{x^{\text{agv}}(t), L - x^{\text{agv}}(t), y^{\text{agv}}(t), W - y^{\text{agv}}(t)\right\}.$$

根据前文所述的AGV是否接近边界的判断和转向逻辑，以及式(8)对AGV的角速度和线速度进行调整。并根据式(9)计算风险系数K，再结合式(10)对DDPG的Actor网络推理输出的原始AGV线速度进行调整。运行时的动作预演直接预判动态障碍物的轨迹，进一步提升了AGV在复杂动态环境下的导航性能。

3　仿真实验及分析

为验证基于改进型DDPG的AGV算法的性能，首先测试并对比了3种DRL算法的训练效果，然后在不同障碍物数量、尺寸、移动速度下对其进行仿真。同时，将A*算法(Liu et al.，2024)、DQN(Hafiz，2022)、DDPG(Lillicrap et al.，2015)作为对比算法，考察AGV的成功率、超时率、碰撞率和耗时指标。

3.1　仿真平台和场景参数

仿真平台环境是64位操作系统Microsoft Windows11 25H2，硬件配置为Intel Core(TM) i7-1165G7@2.80 GHz、24 GB系统内存。仿真环境参数见表1，AGV和障碍物的初始坐标随机选取。

表1　仿真环境参数

Table 1　Simulation environment parameters

参数意义	参数符号	数值
场景范围(长，宽)/m	L, W	50, 50
AGV半径/m	R^{agv}	0.5
边界预警距离/m	d_{bou}	5
缓冲距离/m	d_{buf}	1
速度增益系数	ε	1.5
目的地区域圆半径/m	d^{dest}	1
AGV速度最大值/(m·s^{-1})	$v_{\max}^{\text{agv}}$	2
AGV角速度最大值/(rad·s^{-1})	$\omega_{\max}^{\text{agv}}$	π
时间片段/s	Δt	1

3.2　改进型DDPG算法的训练

本研究在pytorch框架下，采用Python编程实现改进型DDPG算法，其中Pytorch版本为2.9.1+CPU，Python版本为3.12.10。改进型DDPG训练所需参数如表2所示。

对比改进型DDPG、常规DDPG和DQN算法的训练收敛情况，结果如图4所示。图中展示了训练1 000轮的情况下，3种算法的步均奖励情况。从图4中可观察到，常规DDPG算法在350轮之后才趋于收敛，DQN算法在200轮之后趋于收敛，而本文所提的改进型DDPG算法在150轮之后就趋于收敛，收敛速度明显快于对比算法。此外，常规DDPG的

表2 改进型DDPG训练所需参数

Table 2 Training parameters of improved DDPG

参数意义	符号或变量	数值
Actor网络学习率	α	1×10^{-4}
Critic网络学习率	β	1×10^{-3}
折扣因子	γ	0.99
新鲜度时间窗	$T_{\max}$	10 000
风险系数的权重	λ_K	0.1
经验新鲜度权重	λ_F	0.9
批次大小	N	64
训练轮数	TRAIN_EPISODES	1 000
每轮最大步数	MAX_STEPS	100
优化器	Adam	
探索噪声标准差	NOISE_SIGMA	0.1
噪声衰减系数	NOISE_DECAY	0.999
软更新系数	η	0.005

步均奖励收敛于40左右，DQN的步均奖励收敛于[20, 40]，而改进型DDPG的步均奖励收敛于[60, 80]，明显高于对比算法。即改进型DDPG算法收敛速度更快、效果更优。DQN算法的动作空间离散，即AGV的速度和角速度只能取有限离散值，因此对AGV的运动控制不够灵活，时常出现“无法对准”目的地、避障时绕远或避让距离不够等情况，导致奖励值偏低。而常规DDPG算法虽能输出连续动作，但由于AGV导航涉及高维状态空间，在有限的轮次内，DDPG难以学到有效策略。而改进型DDPG采用基于环境风险的自适应动作调整、多因素加权奖励函数、基于风险系数-经验新鲜度双因素的采样策略，提高了输出动作和训练样本的质量，进而提升了训练效率、改善了学习效果。

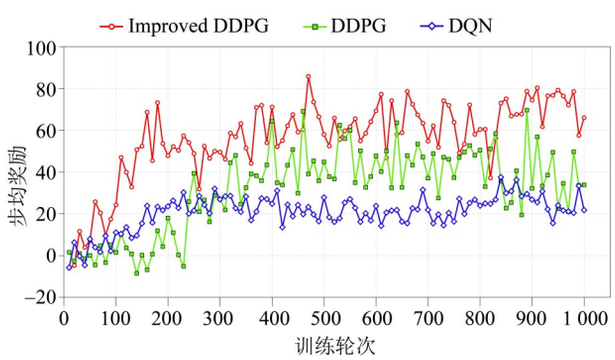

图4 改进型DDPG、DDPG和DQN的训练收敛曲线

Fig. 4 Training convergence curves of improved DDPG, DDPG and DQN

3.3 AGV导航的性能

3.3.1 不同障碍物数量下的导航性能 为考察基于改进型DDPG的AGV导航算法性能，在不同障碍物数量下测试其成功率、超时率、碰撞率及耗时指标。仿真结果见图5~8，其中超时阈值设为100 s。

本文聚焦动态环境下的AGV导航问题，故仿真时设置了动态障碍物。由于A*算法不具备动态避障能力，因此其导航成功率会随着动态障碍物数量的增加而明显下降，相应的碰撞率明显升高。DQN算法仅能输出离散动作，灵活性较差，易出现碰撞、绕远等情况，故而其超时率和碰撞率较高，且随着障碍物数量的增加呈现升高趋势，其成功率亦偏低。常规DDPG算法能输出连续动作，总体上讲，其超时率低于DQN，成功率高于DQN，碰撞率与DQN接近。本文所提的改进型DDPG由于训练充分，能输出高质量连续动作，因此超时率和碰撞率均处于低位，成功率明显高于对比算法。

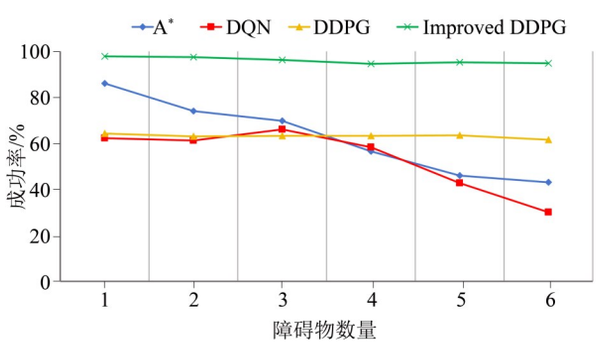

图5 不同障碍物数量下的成功率

Fig. 5 Success rate under different obstacle numbers

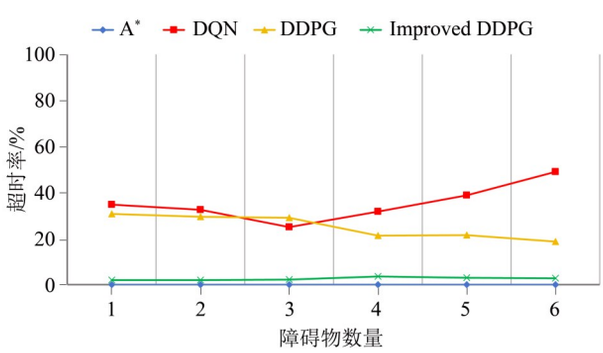

图6 不同障碍物数量下的超时率

Fig. 6 Timeout rate under different obstacle numbers

仿真实验统计了各算法在未碰撞（含未越界）情况下的导航耗时（见图8），因为碰撞意味着导航失败，即便耗时很短（即短时间内发生碰撞或越界），也不能视为节省时间。A*算法规划路径的依据是初始时的环境状态，而忽略后续环境的动态变

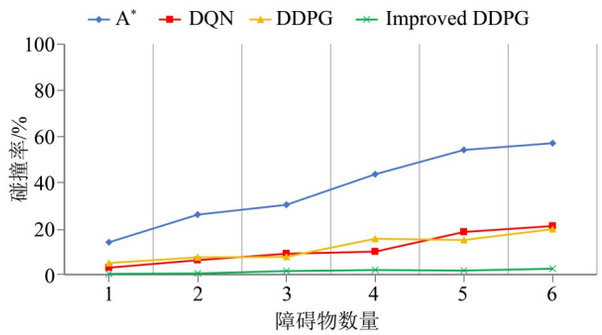

图7　不同障碍物数量下的碰撞率

Fig. 7　Collision rate under different obstacle numbers

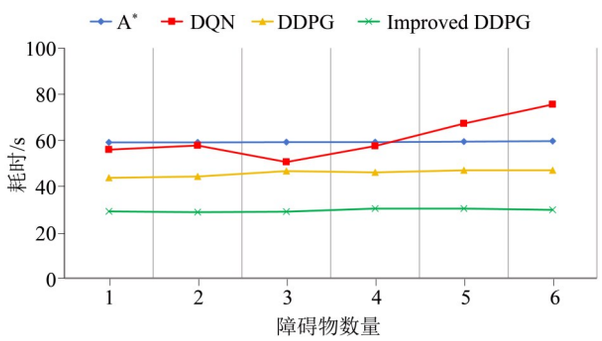

图8　不同障碍物数量下的耗时

Fig. 8　Time consumption under different obstacle numbers

化，无法在AGV行进过程中探索出更优路径，因此其耗时虽较稳定但也偏高。随着障碍物数量的增加，环境变得愈发复杂，DQN的耗时呈现上升趋势。DDPG算法虽然比A*算法和DQN更节省时间，但依然逊于本文的改进型DDPG算法。

3.3.2　不同障碍物尺寸下的导航性能　本研究继续在不同障碍物尺寸的情况下开展仿真实验。虽然实际场景中障碍物的外形各异甚至不规则，但不失一般性，仿真时可用障碍物的外接圆表示障碍物的区域范围，用圆的直径表示障碍物的尺寸。各算法的成功率、超时率、碰撞率、耗时分别见图9~12。

A*算法不能动态避障，其AGV沿初始规划路径

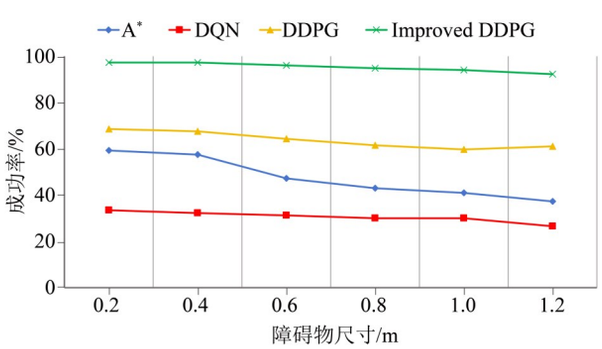

图9　不同障碍物尺寸下的成功率

Fig. 9　Success rate under different obstacle sizes

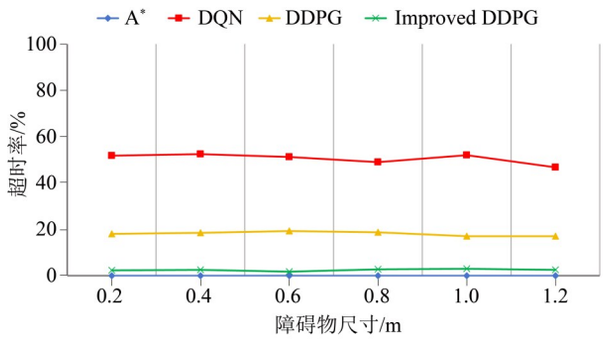

图10　不同障碍物尺寸下的超时率

Fig. 10　Timeout rate under different obstacle sizes

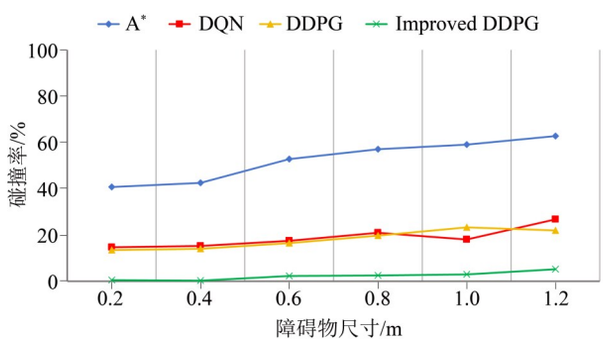

图11　不同障碍物尺寸下的碰撞率

Fig. 11　Collision rate under different obstacle sizes

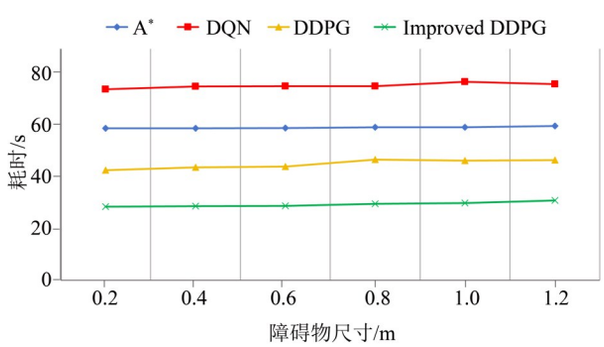

图12　不同障碍物尺寸下的耗时

Fig. 12　Time consumption under different obstacle sizes

行进，若遇到障碍物则碰撞；显然障碍物尺寸越大，碰撞概率越高，相应的成功率越低。DQN、DDPG、改进型DDPG可动态避障，能在一定程度上缓解障碍物尺寸增加带来的不利影响，其中改进型DDPG训练更充分、成功率最高、碰撞率最低。DQN、DDPG虽能处理动态避障问题，但耗时较长，超时率不低。改进型DDPG在奖励函数的设计中综合考虑了避障、防越界、尽快到达目的地等多个优化目标，且抽取了高质量的经验样本进行训练，仿真实验中其碰撞率、超时率、耗时均处于低位。

3.3.3　不同障碍物速度下的导航性能　考虑到AGV工作场景中的移动障碍物大多是行人、移动机

器人等，所以障碍物速度设置在 2 m/s 以内，各算法的成功率、超时率、碰撞率、耗时分别见图 13~16。

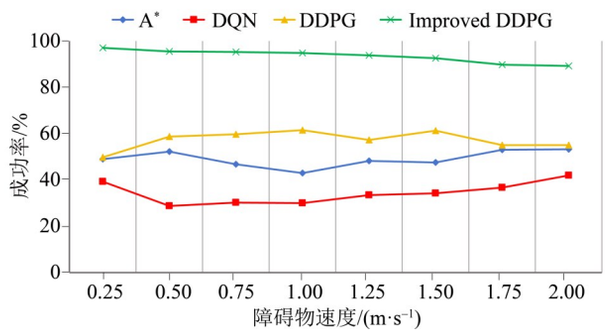

图 13　不同障碍物速度下的成功率

Fig. 13　Success rate under different obstacle speeds

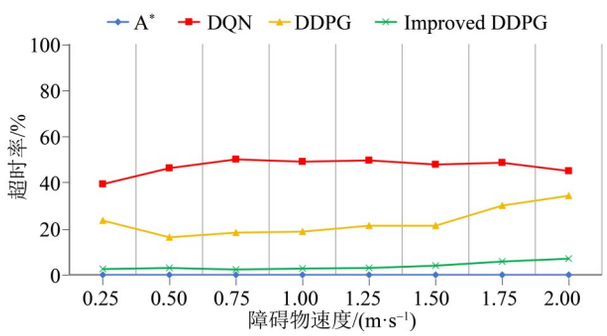

图 14　不同障碍物速度下的超时率

Fig. 14　Timeout rate under different obstacle speeds

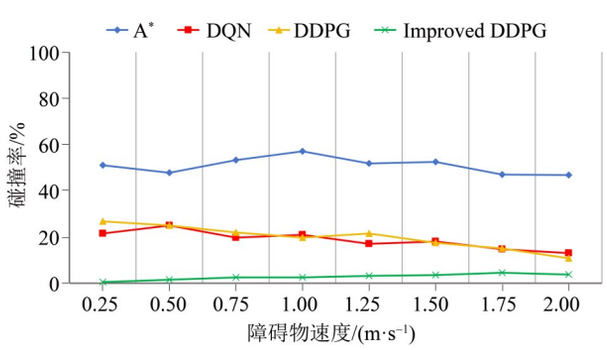

图 15　不同障碍物速度下的碰撞率

Fig. 15　Collision rate under different obstacle speeds

障碍物速度增加，未必给导航算法带来更多负面影响，因为还需要考虑障碍物的运动方向。若障碍物向靠近 AGV 的方向运动，速度越快则对 AGV 越不利；若障碍物向远离 AGV 的方向运动，速度越快则 AGV 越容易避免追尾撞击。从图 13~16 中观察到，障碍物速度增加未必会导致成功率下降以及超时率、碰撞率、耗时增加。即便如此，改进型 DDPG 的成功率仍然最高，碰撞率和耗时最低，超时率也处于低位；A*算法虽然超时率处于低位，但是

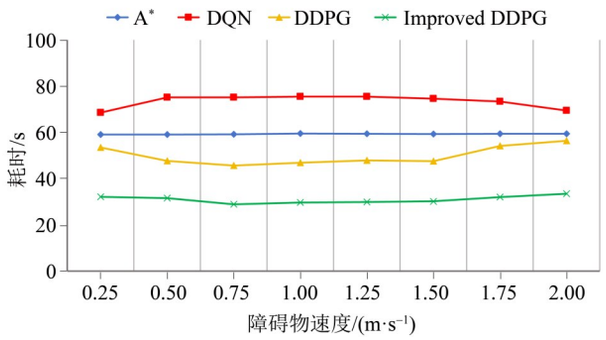

图 16　不同障碍物速度下的耗时

Fig. 16　Time consumption under different obstacle speeds

碰撞率和耗时较高，成功率较低；DQN 算法只能输出有限的离散动作，灵活性受限，因此其超时率最高、耗时最长，且存在一定的碰撞率，导致成功率最低；常规 DDPG 算法的碰撞率和 DQN 接近，但超时率和耗时低于 DQN，且其超时率和耗时高于改进型 DDPG，成功率也低于改进型 DDPG。

3.4　AGV 导航的轨迹

本文开展了大量实测，并选取不同障碍物数量、尺寸、速度组合条件下的 8 组导航数据，将其轨迹绘制于图 17 中。从图中可观察到，多个障碍物无规律地在场景不同区域移动，而 AGV 在改进型 DDPG 导航算法的指引下，从起点经过障碍物区域到达终点，避开了所有障碍物，未越界；且均能在 25 步内到达目的地，体现了改进型 DDPG 导航算法的优越性。

4　结　语

本研究提出了基于改进型 DDPG 的 AGV 智能导航算法，其改进包括 4 个方面：1）在 Actor 网络输出动作时，采用基于环境风险的自适应探索策略，输出优质探索动作，进而缩短训练周期并提升训练质量；2）设计了以避障、防越界、尽早到达目的地为目标的加权奖励函数，以提升导航算法对复杂环境的适应性；3）基于风险系数-经验新鲜度双因素抽取经验，提升训练样本的质量，加快收敛速度；4）在运行阶段通过动作预演弥补动态障碍物轨迹预判方面的不足，进一步提升导航的性能。

从 3 种 DRL 算法的训练收敛曲线可知，改进型 DDPG 收敛速度最快，收敛后的步均奖励值最高。不同障碍物数量、尺寸、移动速度条件下的仿真实验显示：相较于对比算法，基于改进型

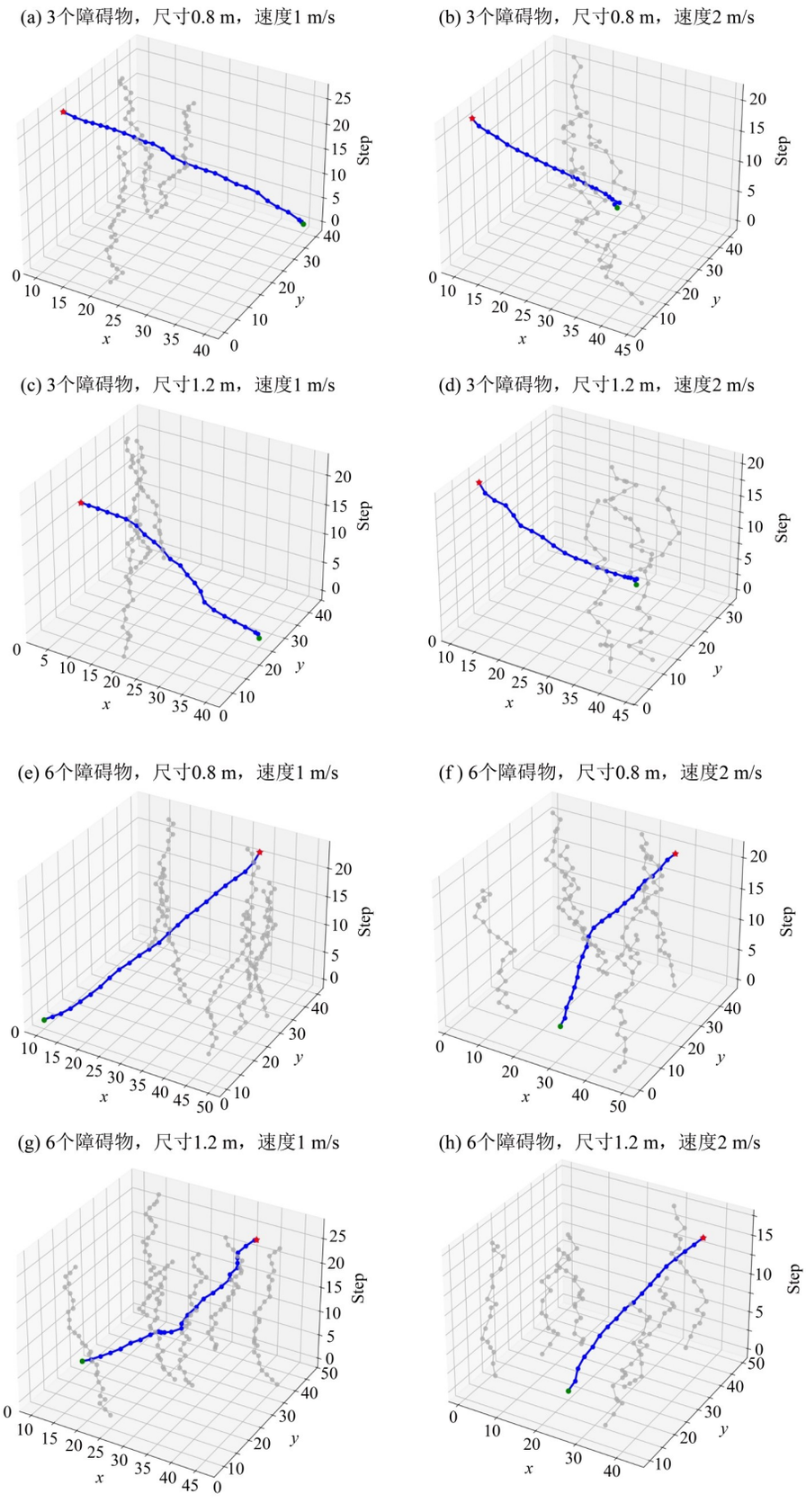

图 17　基于改进型DDPG的AGV导航轨迹图

Fig. 17　AGV navigation trajectory with improved DDPG

DDPG的AGV算法成功率最高，耗时最短、平均碰撞率最低，平均超时率也处于低位，这验证了该算法的优越性能，体现出其具有良好的有效性与鲁棒性。

参考文献：

王贺，许佳宁，闫广宇，2025. 基于深度强化学习的AGV行人避让策略研究[J]. 系统仿真学报，37(3)：595-606.

王唯鉴，2023.AGV动态避障及任务调度深度强化学习研究[D]. 北京：机械科学研究总院.

Arulkumaran K，Deisenroth M P，Brundage M，et al，2017. Deep reinforcement learning：A brief survey[J]. IEEE Signal Process Mag，34(6)：26-38.

Chen X Q，Liu S H，Li C F，et al，2023. AGV path planning and optimization with deep reinforcement learning model[C]//7th International Conference on Transportation Information and Safety. Xi'an，China：1859-1863.

Gao T H，Chen B C，Mi Q W，2022. A survey of Markov model in reinforcement learning[C]//2022 International Conference on Artificial Intelligence in Information and Communication. Jeju Island，South Korea：284-287.

Hafiz A M，2022. A survey of deep Q-networks used for reinforcement learning：State of the art[C]//Intelligent Communication Technologies and Virtual Mobile Networks：Proceedings of ICICV 2022. Singapore：393-402.

Lillicrap T P，Hunt J J，Pritzel A，et al，2019. Continuous control with deep reinforcement learning[PP/OL].[2026-04-03].https://doi.org/10.48550/arXiv.1509.02971.

Liu N，Ma C Y，Hu Z H，et al，2024. Workshop AGV path planning based on improved A* algorithm[J].Math Biosci Eng，21(2)：2137-2162.

Shen G C，Ma R，Tang Z，et al，2021. A deep reinforcement learning algorithm for warehousing multi-AGV path planning[C]//International Conference on Networking，Communications and Information Technology. Beijing，China：421-429.

Sutton R S，Barto A G，2018. Reinforcement learning：An introduction[M]. Cambridge：The MIT Press.

Wang X，Wang S，Liang X X，et al，2024. Deep reinforcement learning：A survey[J]. IEEE Trans Neural Netw Learn Syst，35(4)：5064-5078.

Wu H，2025. Research on AGV path planning algorithm integrating adaptive A* and improved APF algorithm[C]//8th International Conference on Advanced Algorithms and Control Engineering. Shanghai，China：764-769.

Zhang B，Zhu M W，Lin C H，et al，2022. Research on AGV map building and positioning based on SLAM technology[C]//5th International Conference on Automation，Electronics and Electrical Engineering. Shenyang，China：707-713.

（责任编辑　王海蓉）